Claims

What is claimed is:

1. A method for implementing atomic data tracing in a processor system including an auxiliary processor unit coupled to a central processor unit (CPU), using the auxiliary processor unit (APU) to perform the steps of:

identifying a trace instruction;

signaling the CPU with a pipeline stall signal for stalling the CPU;

checking for an enabled trace engine for said trace instruction,

writing trace data into a trace buffer responsive to an identified enabled trace engine for said trace instruction; and

signaling the CPU with an op done signal for allowing the CPU to continue with instruction processing.

2. A method for implementing atomic data tracing as recited in claim 1 wherein the step of writing trace data into said trace buffer includes the step of utilizing a set of device control registers (DCRs) accessible by the APU to determine where to write said trace data in said trace buffer.

3. A method for implementing atomic data tracing as recited in claim 2 wherein said set of device control registers (DCRs) include a trace buffer pointer register for storing a base address of said trace buffer and an offset indicating a current trace buffer entry.

4. A method for implementing atomic data tracing as recited in claim 3 wherein the step of writing trace data into said trace buffer includes updating said offset into said current trace buffer entry of said trace buffer pointer register.

5. A method for implementing atomic data tracing as recited in claim 3 wherein said set of device control registers (DCRs) include a base address mask register storing a mask indicating which bits in said trace buffer pointer register hold said base address and which hold said offset; said base address mask register used to determine a wrap point of said trace buffer.

6. A method for implementing atomic data tracing as recited in claim 3 wherein said set of device control registers (DCRs) include a control register storing an enabled bit indicating whether or not said trace engine for said trace instruction is enabled, said enabled bit being used for allowing data tracing to be turned on and off on the fly.

7. A method for implementing atomic data tracing as recited in claim 6 wherein said control register includes a time stamp value indicating whether or not a time stamp should be traced; and wherein the step of writing trace data into said trace buffer includes writing a time stamp with said trace data responsive to said control register time stamp value.

8. A method for implementing atomic data tracing as recited in claim 6 wherein said control register includes a number field indicating a number of bytes to be traced; and wherein the step of writing trace data into said trace buffer includes updating said offset into said current trace buffer entry of said trace buffer pointer register by said number of bytes.

9. A method for implementing atomic data tracing as recited in claim 3 wherein said trace instruction includes a number field indicating a number of bytes to be traced; and wherein the step of writing trace data into said trace buffer includes updating said offset into said current trace buffer entry of said trace buffer pointer register by said number of bytes.

10. A method for implementing atomic data tracing as recited in claim 3 responsive to identifying no enabled trace engine for said trace instruction, signaling the CPU with said op done signal for allowing the CPU to continue with instruction processing without writing trace data.

11. Apparatus for implementing atomic data tracing in a processor system including an auxiliary processor unit (APU) coupled to a central processor unit (CPU), said apparatus comprising:

a trace engine; said trace engine including a set of device control registers (DCRs) accessible by the APU, and a trace buffer;

a trace instruction; said trace instruction including encoded first and second general purpose registers (GPRs), said first GPR containing an index to said trace engine DCRs and said second GPR indicating a first GPR containing data to be written into a current trace entry in said trace buffer;

the APU processes said trace instruction performing the steps of signaling the CPU with a pipeline stall signal for stalling the CPU; responsive to identifying an enabled trace engine for said trace instruction, writing trace data into said trace buffer; and signaling the CPU with an op done signal for allowing the CPU to continue with instruction processing.

12. Apparatus for implementing atomic data tracing in a processor system as recited in claim 11 wherein said set of device control registers (DCRs) include a trace buffer pointer register for storing a base address of said trace buffer and an offset into a current trace buffer entry; a base address mask register storing a mask indicating which bits in said trace buffer pointer register hold said base address and which hold said offset; said base address mask register used to determine a wrap point of said trace buffer; and a control register storing an enabled bit indicating whether or not said trace engine for said trace instruction is enabled, said enabled bit being used for allowing data tracing to be turned on and off on the fly.

13. Apparatus for implementing atomic data tracing in a processor system as recited in claim 12 wherein the APU updates said offset to said current trace buffer entry of said trace buffer pointer register for each trace data entry written to said trace buffer.

14. Apparatus for implementing atomic data tracing in a processor system as recited in claim 12 wherein said control register includes a time stamp value indicating whether or not a time stamp should be traced; and the APU writes a time stamp with said trace data responsive to said control register time stamp value.

15. A computer program product for implementing atomic data tracing in a processor system including an auxiliary processor unit (APU) coupled to a central processor unit (CPU), said computer program product including instructions executed by the processor system to cause the processor system to perform the steps of:

defining a trace buffer and a set of device control registers (DCRs) accessible by the APU to create a trace engine;

providing a trace instruction; said trace instruction including encoded first and second general purpose registers (GPRs), said first GPR containing an index to said trace engine DCRs and said second GPR indicating a first GPR containing data to be written into a current trace entry in said trace buffer; and

utilizing the APU for processing said trace instruction by performing the steps of: signaling the CPU with a pipeline stall signal for stalling the CPU; utilizing said set of DCRs, checking for an enabled trace engine for said trace instruction; and responsive to identifying an enabled trace engine for said trace instruction, writing trace data into said trace buffer; and signaling the CPU with an op done signal for allowing the CPU to continue with instruction processing.

16. A computer program product for implementing atomic data tracing as recited in claim 15 wherein the step of defining a set of device control registers (DCRs) accessible by the APU to create a trace engine includes the steps of defining said set of DCRs including a trace buffer pointer register for storing a base address of said trace buffer and an offset into a current trace buffer entry; a base address mask register storing a mask indicating which bits in said trace buffer pointer register hold said base address and which hold said offset; said base address mask register used to determine a wrap point of said trace buffer; and a control register storing an enabled bit indicating whether or not said trace engine for said trace instruction is enabled, said enabled bit being used for allowing data tracing to be turned on and off on the fly.

17. A computer program product for implementing atomic data tracing as recited in claim 16 wherein the step of writing trace data into said trace buffer includes updating said offset to said current trace buffer entry of said trace buffer pointer register for each trace data entry written to said trace buffer.

18. A computer program product for implementing atomic data tracing as recited in claim 16 wherein the step of writing trace data into said trace buffer includes checking a time stamp value in said control register indicating whether or not a time stamp should be traced; and the APU writes a time stamp with said trace data responsive to said control register time stamp value.